Interconnection Architecture for Spontaneous Edge Networks

Vincent Untz
Advisors: Martin Heusse, Andrzej Duda
LSR-IMAG Laboratory
Grenoble, France
Email: vincent.untz@imag.fr

I. Aims and Scope

We consider the problem of *spontaneous edge networks* illustrated in Figure 1. With this term we designate networks that interconnect hosts by means of different physical and link layer technologies and in which all or some of hosts are organized as a multihop ad hoc network. A spontaneous network can be connected to the global Internet or form an isolated group of hosts with internal connectivity. Such networks are becoming widespread with the advent of various communicating devices at home and in offices as well as with the development of pervasive devices connected via different types of networks and integrated within the physical world.

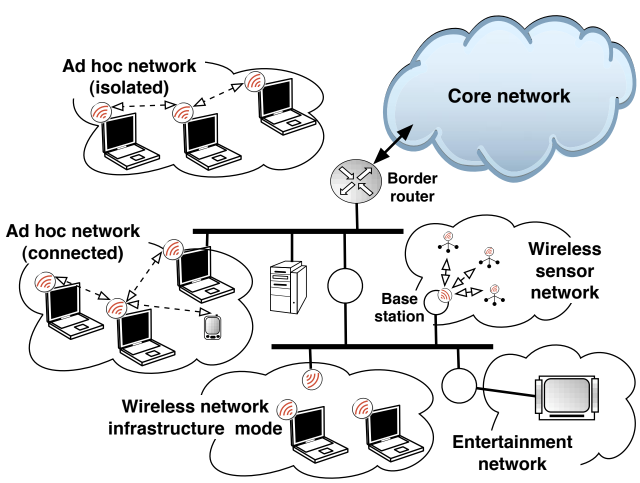

Fig. 1. Spontaneous edge network

The interconnection of hosts in home and office networks has been analyzed by Krishnamurthy *et al.* [1]. They have identified the main requirements of wireless home networks: autoconfiguration and meeting performance objectives by using multi-hop wireless technologies. However, no solutions have been proposed.

We share their analysis and our main goal is to define an interconnection architecture that fulfills these requirements. Moreover, the architecture should support TCP/IP applications without configuration or other technical effort from the user. The applications should work when a spontaneous network is connected to the global Internet or not, as well as when its topology changes due to host mobility or switching network interfaces. Moreover, the architecture needs to support different kinds of applications, especially those that generate time dependent traffic.

To address the issues raised by spontaneous edge networks, we have designed and implemented Lilith, a prototype of an interconnection node for spontaneous edge networks. Because of space limitation, we cannot describe here the architecture of Lilith. The motivations for Lilith and its design can be found elsewhere [2].

II. Layer 2.5 Spontaneous Networking

Lilith is based on existing standard technologies (MPLS, on demand ad hoc routing) which, glued together, present valuable features for spontaneous networking. The idea is to organize a spontaneous network as a single IP subnetwork (broadcast or scoped multicast packets propagate to all hosts), to interconnect various links with MPLS and to manage LSPs (Label Switched Paths) on demand with a reactive ad hoc routing protocol.

The main principles of Lilith are:

- we make all the hosts connected via different links appear as one single IP subnet so that configuration protocols can use the subnet broadcast (IPv4) or the scoped multicast (IPv6) for all forms of discovery (addresses, names, services);
- we propose to interconnect hosts at layer 2.5, which enables us to easily integrate an ad hoc routing protocol;
- we build our own 2.5 layer upon MPLS *(Multi Protocol Label Switching)*, the standard 2.5 layer;
- LSPs (Label Switched Paths) transport packets between different links;
- the establishment of LSPs happens on demand and is driven by a reactive ad hoc routing protocol;
- in order to guarantee that all hosts in the network are reachable, we use a multihop routing protocol.

III. QoS Routing Made Easy

We notice that a spontaneous network may carry various types of traffic ranging from low bandwidth sensor information to time sensitive interactive traffic or high bandwidth, low latency home entertainment streams. Hence, the interconnection architecture needs to support some QoS aspects such as traffic classification and prioritization. It is also desirable to take advantage of multiple available paths and adjust to varying traffic conditions and changing topology.

Using on demand label switching is a first step towards handling the network dynamics: not only the topology of a spontaneous network may change over time, but also the network may experience important load variations. Since Lilith contains an optimization module that tries to explicitly find alternate routes for existing active destinations, our approach can cope with these changes by detecting broken or better paths.

One step further will be to take into account some QoS parameters in route discovery, for instance give more weight to routes that use more wired hops or feature wireless links with better quality. We can then explore different types of QoS based routing, because all possibilities are being opened by the connection oriented nature of Lilith. MPLS will enable us to easily investigate flow-based load balancing.

We are working on more performance experiments to measure the benefits of the various QoS approaches.

IV. Optimized Broadcast Handling

Since we consider a spontaneous network as a single IP subnet, our architecture propagates layer 3 broadcasts or scoped multicasts to all hosts. They can then be used for all configuration protocols. In particular, when the network is isolated from the global Internet, hosts can acquire addresses based on Auto IPv4 or stateless IPv6 configuration and use mDNS or LLMNR for name resolution. When connected to the border router, they benefit from a DHCP service to learn the routable prefix. Moreover, protocols such as UPnP, SLP, or JINI can readily be used for service discovery.

We also note that more and more popular applications use layer 3 broadcasts for service discovery. This observation and the fact that autoconfiguration is one of our main goals lead to the conclusion that layer 3 broadcasts will be used, and maybe, abused: having an important number of layer 3 broadcasts in a spontaneous edge network may have non negligeable impact on performance.

For this reason, we consider different approaches to optimize layer 3 broadcasts. The first approach is to not rely on simple flooding, but use a more elaborate approach like defining multipoint relays.

However, Lilith enables us to take another approach we want to explore first: as layer 3 broadcast packets need to reach all hosts in a spontaneous edge network, one task of Lilith is to propagate them by encapsulation in a specific Lilith frame forwarded to all hosts using the same mechanism as for dynamic route discovery. This enables us to distinguish between unicast and broadcast packets to handle the latters in a specific way. For example, instead of propagating each broadcast packet in the network as soon as it is created, we can wait for some time and concatenate the broadcast packets created over the time interval to send them all at once in a Lilith-specific frame. Delaying broadcast packets for a reasonable amount of time should not have an impact on applications, since applications do not assume broadcasts to be reliable anyway. This approach leads to performance gains because the network resources are used more efficiently. Moreover, it would limit broadcast storms that may often happen in ad hoc networks when a topology change occurs.

V. Conclusion

We have designed and implemented Lilith to solve issues specific to spontaneous edge networks. It provides a platform that enables us to perform more tests and experiments with this kind of networks.

The crucial benefit of Lilith comes from the fact that it forms a 2.5 layer. We believe that this approach will benefit to the end user because it will make the deployment of communicating electronic equipments easier: no advanced technical knowledge is necessary, which is a usually forgotten condition to make ubiquitous computing happen for everyone.

References

[1] L. Krishnamurthy, S. Conner, M. Yarvis, J. Chhabra, C. Ellison, C. Brabenac and E. Tsui *Meeting the Demands of the Digital Home with High-Speed Multi-Hop Wireless Networks*. In *Intel Technology Journal*, vol. 6, issue 4. 2002.

[2] V. Untz, M. Heusse, F. Rousseau, and A. Duda, *On Demand Label Switching for Spontaneous Edge Networks*. In *Proc. SIGCOMM Workshop on Future Directions in Network Architectures*. Portland, USA, August 2004.